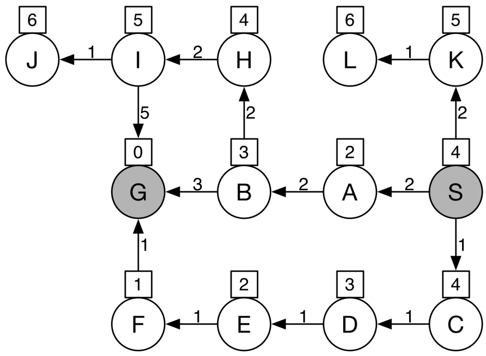
Inteligência Artificial – 2017/18

28/Outubro/17 — 14:30 — Duração: 2h

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

1º Teste
– Com consulta limitada –

I) [7val] Considere o seguinte grafo de estados de um problema de procura. Os valores apresentados nos arcos correspondem ao custo do operador (ação) respectivo, enquanto os valores nos rectângulos correspondem ao valor de uma heurística (estimativa do custo de chegar desse nó ao nó objectivo (G)). Não se representam os nomes dos operadores, correspondendo cada arco a um operador distinto. Assuma que os sucessores de um nó são sempre gerados por ordem alfabética do nome do nó (ou seja, por exemplo, dos sucessores de S, o primeiro a ser gerado é o A, o segundo é o C e o terceiro é o K). A ordem alfabética deverá também ser utilizada sempre que for necessário desempatar entre nós. Pretende-se encontrar um caminho desde o nó S até ao nó G.



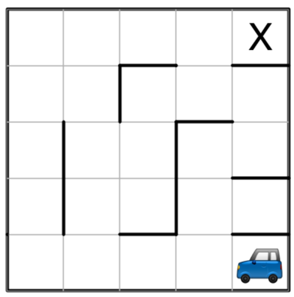
a) Considere os algoritmos de procura em largura primeiro , procura de custo uniforme e procura sôfrega. Para cada um dos algoritmos, indicar:
   - Qual o caminho encontrado pelo algoritmo
   - Quais os nós que são expandidos

b) Caracterize a heurística quanto à admissibilidade e consistência, justificando a sua resposta.

c) Ilustre como se comporta o algoritmo de procura A* em grafos na resolução deste problema. Use a versão do algoritmo que garante a obtenção da solução óptima dadas as características da heurística. Deve explicitar os conteúdos das estruturas de dados auxiliares ao longo das iterações do algoritmo, colocando entre parêntesis o valor da função de avaliação para cada nó na lista. Indique o caminho encontrado pelo algoritmo, assim como o seu custo.

d) Para cada algoritmo da alínea a) que não tenha encontrado o caminho óptimo, explique porquê.

II) [3val] Considere um agente (carro) que pretende atingir a casa de saída (X) de um labirinto como o ilustrado na imagem ao lado. Em cada momento, o agente está virado para uma direção $d \in \{N, S, E, W\}$. Com uma única ação, o agente pode mover-se em frente a uma velocidade ajustável, ou mudar de direção. As ações de mudança de direção são *direita* e *esquerda*, que alteram a direção do agente em 90º. Mudanças de direção só são permitidas quando a velocidade é zero, deixando o carro com uma velocidade de zero, virado na nova direção. As ações de movimentação são *acelerar* e *travar*. Acelerar aumenta a velocidade em 1 e travar reduz a velocidade em 1; em ambos os casos, o agente avança o numero de casas igual á sua nova velocidade. Qualquer ação que leve o agente a embater numa parede é ilegal. Qualquer ação que reduza a velocidade abaixo de 0 ou acima de v_{max} é também ilegal. O objetivo do agente é encontrar uma sequência de ações que o deixe parado na casa de saída, usando o menor numero de ações.



Como exemplo, se o carro estivesse inicialmente parado, poderia *acelerar*, avançando uma casa na direção W, após o qual poderia *acelerar* novamente avançando duas casas, e depois *travar*, avançando uma casa, e *travar* novamente de modo a parar. Em seguida, poderia rodar à *direita*, repetindo a sequencia de ações anteriores mais duas vezes até chegar ao objetivo.

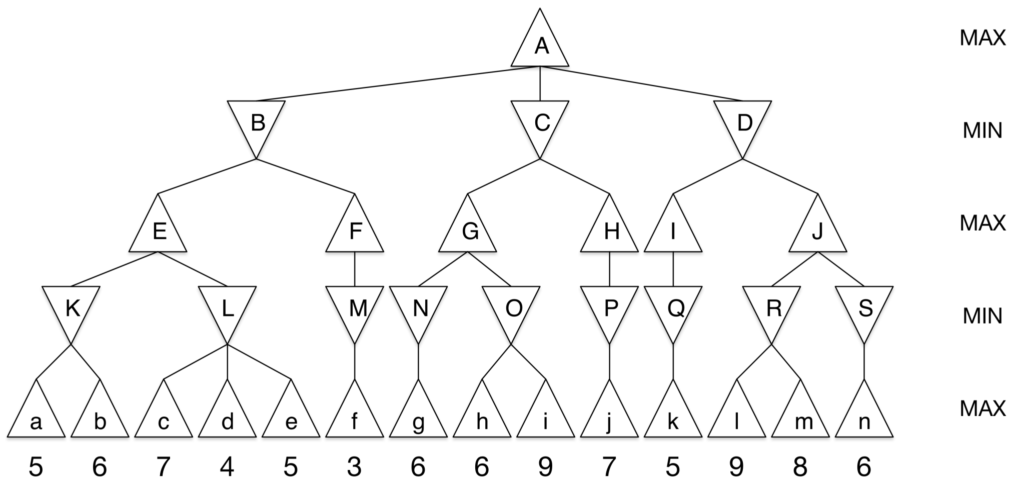
a) Se a grelha for M por N, qual o tamanho do espaço de estados? Justifique a sua resposta. (pode assumir que todas as configurações são atingíveis a partir do estado inicial.
b) Qual o factor de ramificação máximo? Justifique a sua resposta. (pode assumir que as acções ilegais não são devolvidas pela função sucessor)
c) A distância de Manhattan entre o agente e a casa de saída é uma heurística admissível? Justifique a sua resposta.

III) [3val] Sejam X, Y e Z variáveis inteiras, sujeitas às seguintes restrições X >= Y + 3 e Y >= 7 – X – Z . Sabe-se que a variável X pode tomar os valores {5,6,7}, a variável Y os valores {0,1} e a variável Z os valores {0,1,2}. Considere o algoritmo de procura com retrocesso com atribuição em primeiro lugar da variável mais constrangida e desempate pelo número de restrições. Indique a dimensão do espaço de procura e desenhe a árvore de procura até à obtenção de uma solução (explicitando-a), assumindo que os valores das variáveis são ordenados de forma crescente.

IV) [3val] Verifique, usando o algoritmo DPLL, se a seguinte fórmula de lógica proposicional é satisfazível:

$$(A \Rightarrow B) \wedge (A \wedge C \Rightarrow D) \wedge (C \Leftrightarrow B) \wedge (\neg C \vee D) \wedge \big(\neg(C \wedge D)\big)$$

V) [4val] Considere a árvore de jogo de dois jogadores (MAX e MIN), onde os valores nas folhas são estimativas do ganho para MAX a partir desse estado.
a) Calcule o valor minimax de cada nó (não terminal) e indique qual o movimento que MAX deverá escolher.
b) Assumindo que a árvore é percorrida da esquerda para a direita, quais os nós que nunca chegariam a ser visitados pelo algoritmo de busca α−β. Justifique sucintamente.



VI) [Bónus: até 1val] Pretende-se minimizar a função f(x)=(x-10)(x-4) recorrendo ao algoritmo trepa-colinas em que x toma valores inteiros entre 0 e 10. Considere as vizinhanças de um estado obtidas por variação (incremento ou decremento) de uma unidade de x. Indique a sequência de estados obtida pelo algoritmo a partir de um estado inicial à sua escolha e qual o respectivo valor mínimo obtido para a função.

VII) [Bónus: até 1val] O Anacleto defende que: Em geral, se nos for dado um problema de pesquisa para o qual sabemos: i) o valor da heurística para todos os estados; ii) que a heurística é admissível; iii) a solução encontrada pelo A*; e iv) o conteúdo da fronteira e explorados da execução do A*, então é possível determinar de forma unívoca o custo de cada operador (ação) do problema. O Anacleto está correto? Se sim, então esboce uma demonstração. Caso contrário, apresente um exemplo que mostre que o Anacleto está errado.

28/Out/2017 – 14:30 **Inteligência Artificial – Folha de Respostas** **DI/FCT/UNL, 1º Teste, 2017/18, Duração: 2h** **Teste:** addbadcb

Nome:	**Número:**

I.a) Algoritmo	**Solução**	**Expandidos**
largura primeiro	S→A→B→ G	S(0),A(2),C(1),K(2),B(4) (ver. optimizada) ou S(0),A(2),C(1),K(2),B(4),D(2),L(3)
custo uniforme	S→C→D→E→F→G	S(0),C(1),A(2),D(2),K(2),E(3),L(3),B(4),F(4)
sôfrega	S→A→B→ G	S(4),A(2),B(3)

I.b)
Admissível? **Sim** Consistente? **Sim**
Justificação: É consistente pois para quaisquer dois estados n1, n2, ligados por um arco com custo c(n1,n2), temos que h(n1)≤c(n1,n2)+h(n2). Por exemplo, se n1=S e n2=A, temos que h(S)=4, c(S,A)=2 e h(A)=2, logo, h(S) ≤c(S,A)+h(A). Se n1=S e n2=K, temos que h(S)=4, c(S,K)=2 e h(K)=5, logo, h(S) ≤c(S,A)+h(A). Esta relação verifica-se para todos os nós ligados por um arco, pelo que a heurística é consistente.
É admissível pois toda a heurística consistente é admissível.

I.c)
Solução:S→C→D→E→F→G Custo da Solução: 5

Fronteira:	S(4)	A(4),C(5),K(7)	C(5),B(7),K(7)	D(5),B(7),K(7)	E(5),B(7),K(7)
Explorados:		S(4)	S(4),A(4)	S(4),A(4),C(5)	S(4),A(4),C(5),D(5)

Fronteira (cont.):	F(5),B(7),K(7)	G(5),B(7),K(7)	B(7),K(7)	
Explorados (cont.):	S(4),A(4),C(5),D(5),E(5)	S(4),A(4),C(5),D(5),E(5),F(5)	S(4),A(4),C(5),D(5),E(5),F(5)	

Entre parêntesis indica-se o valor da função de avaliação (custo + heurística).

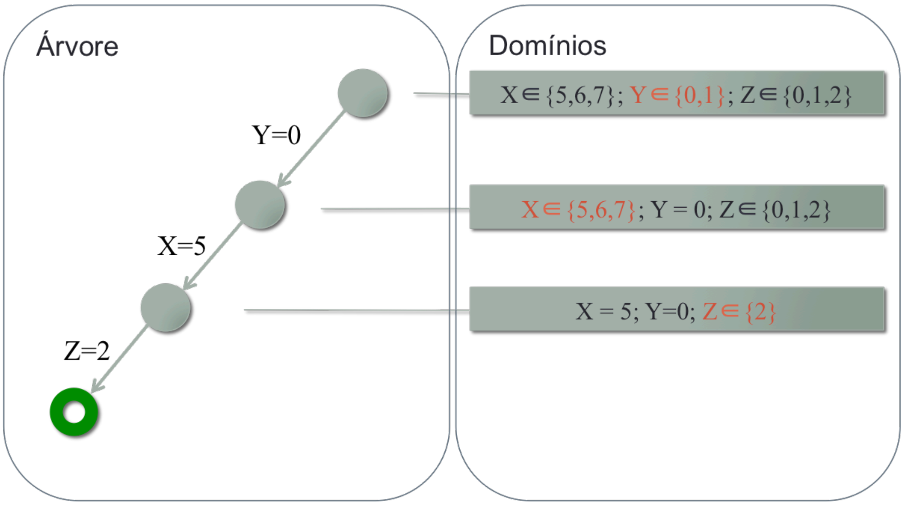
I.d) O algoritmo de procura em largura primeiro tenta encontrar uma solução com o menor número de acções (neste caso encontrando uma com 3 acções), interrompendo a procura sem tentar encontrar uma solução melhor que envolva mais acções (no caso, a óptima tem 5 acções). O algoritmo de procura sôfrega dá mais importância ao valor da heurística do que ao custo para chegar a um estado. Assim, prefere explorar nós correspondentes a estados que estão próximos do objectivo, mesmo que o custo para lá chegar tenha sido elevado, e haja uma melhor solução por outro caminho. Neste caso, por exemplo, quando confrontado com a escolha entre selecionar o nó correspondente ao estado B e o nó correspondente ao estado C, prefere selecionar o nó correspondente a B por parecer estar mais perto do objetivo (h(B)=3), ignorando o facto de existir um custo igual a 4 para chegar a B, ao passo que o valor da heurística em C é h(C)=4, e o custo para chegar a C é de apenas 1.

II.a)
Tamanho? $\mathbf{4 * M * N * (v_{max} + 1)}$
Justificação: A representação do estado é (direcção, x, y, velocidade), e a velocidade pode tomar qualquer valor entre 0 e v_{max}.

II.b)
Factor de Ramificação Máximo? **3**
Justificação: O factor de ramificação máximo acontece quando o agente está estacionário, podendo executar uma de três acções: acelerar, esquerda e direita.

II.c)
Resposta? **Não**
Justificação: A distância de Manhattan não é uma heurística admissível, pois o agente pode mover-se com uma velocidade média superior a 1 (acelerando até v_{max} e travando até 0 quando se aproxima do objectivo), podendo assim atingir o objectivo em menos passos do que o número de casas entre a sua localização e o objectivo. Por exemplo, se o agente estiver a 6 casas do objectivo em linha recta e sem obstáculos, e a sua velocidade for 4, ele chega ao objectivo com velocidade 0 em 4 passos, travando em cada um, pelo que o número de paços seria inferior à distância de Manhattan.

III)
Dimensão do espaço?**18** Solução? X=5, Y=0, Z=2
Árvore:



IV)
Satisfazível? Sim (Um Modelo: A→Falso; B→Falso; C→Falso; D→Verdadeiro)
Verificação: Conversão para a forma clausal:

$$(\neg A \lor B) \land (\neg A \lor \neg C \lor D) \land (C \lor \neg B) \land (\neg C \lor B) \land (\neg C \lor D) \land (\neg C \lor \neg D)$$

Aplicação do DPLL:

$$\begin{array}{c}(\neg A \lor B)\\(\neg A \lor \neg C \lor D)\\(C \lor \neg B)\\(\neg C \lor B)\\(\neg C \lor D)\\(\neg C \lor \neg D)\end{array} \xRightarrow[(\neg A)]{Pure} \begin{array}{c}(C \lor \neg B)\\(\neg C \lor B)\\(\neg C \lor D)\\(\neg C \lor \neg D)\end{array} \xRightarrow[(B)]{Split} \begin{array}{c}(C)\\(\neg C \lor D)\\(\neg C \lor \neg D)\end{array} \xRightarrow[(C)]{Unit} \begin{array}{c}(D)\\(\neg D)\end{array} \xRightarrow[(D)]{Unit} \bot$$

$$\xRightarrow[(\neg B)]{BackTrack} \begin{array}{c}(\neg C)\\(\neg C \lor D)\\(\neg C \lor \neg D)\end{array} \xRightarrow[(\neg C)]{Pure} \top$$

Há outras alternativas válidas...

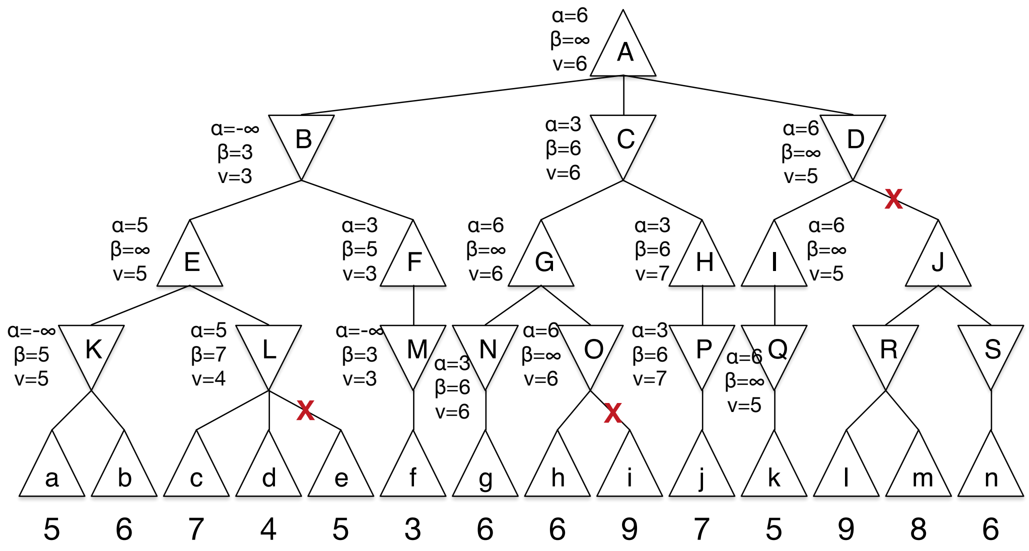
28/Out/2017 – 14:30 **Inteligência Artificial – Folha de Respostas**
DI/FCT/UNL, 1º Teste, 2017/18, Duração: 2h **Teste:** addbadcb

Nome:	**Número:**

V.a)
Minimax: A:**6** B:**3** C:**6** D:**5** E:**5** F:**3** G:**6** H:**7** I:**5** J:**8** K:**5** L:**4** M:**3** N:**6** O:**6** P:**7** Q:**5** R:**8** S:**6**

Movimento. de MAX: **A→C**

b)
Nós não visitados: **e, i, J, R, S, l, m, n**
Justificação: (em vez de apresentar a árvore, poderia ter explicado porque é que cada um dos 3 ramos foi cortado)



VI)
Sequência? **4,5,6,7** ou **9,8,7** ou **0,1,2,3,4,5,6,7** ou **7** ou ... Valor mínimo? **-9**

VII)
Sim/Não? **Não, o Anacleto não está correto.**
Demonstração/Exemplo: Só é possível inferir o custo de uma ação de um estado para outro quando um nó com esse estado é expandido. Como o A*, em geral, deixa parte do espaço por explorar, existem estados para os quais não chega a haver um nó expandido, havendo inclusivamente estados que nunca foram visitados i.e. para os quis nunca chegaram a ser criados nós. O exemplo da pergunta 1c) demonstra-o. No fim da execução do A*, os nós com estados B e K nunca chegaram a ser expandidos pelo que seria impossível determinar os custos das acções a partir desses estados. Igualmente, como os estados L, H, I e J nunca chegaram a ser visitados, também não foram expandidos, pelo que também seria impossível determinar os custos das acções a partir desses estados.